

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND/ODESSA DIVISION**

VIRTAMOVE, CORP.,

Plaintiff,

v.

ORACLE CORP.,

Defendant.

Case No. 7:24-CV-00339-ADA

JURY TRIAL DEMANDED

**DEFENDANT ORACLE CORP.'S MOTION TO DISMISS PLAINTIFF'S
COMPLAINT PURSUANT TO FED. R. CIV. P 12(b)(6)**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Legal Standards.....	1
III.	Argument	3
A.	The '814 Patent Claims Are Invalid For Claiming Ineligible Subject Matter.....	3
1.	Background.....	3
2.	The '814 Patent Is Directed To An Abstract Idea.....	5
3.	The Asserted Claim Adds Nothing Inventive	9
4.	The Remaining Claims Are Not Patent Eligible.....	11
B.	The '058 Patent Claims Are Invalid For Claiming Ineligible Subject Matter.....	12
1.	Background.....	12
2.	The '058 Patent Is Directed To An Abstract Idea.....	14
3.	The Asserted Claim Adds Nothing Inventive	16
4.	The Remaining Claims Are Not Patent Eligible.....	18
IV.	Conclusion	18

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Aatrix Software, Inc. v. Green Shades Software, Inc.</i> , 882 F.3d 1121 (Fed. Cir. 2018).....	2
<i>Affinity Labs of Tex., LLC v. DIRECTV, LLC</i> , 838 F.3d 1253 (Fed. Cir. 2016).....	1
<i>Alice Corp. Pty. Ltd. v. CLS Bank Int’l</i> , 573 U.S. 208 (2014).....	<i>passim</i>
<i>Berkheimer v. HP Inc.</i> , 881 F.3d 1360 (Fed. Cir. 2018).....	7, 11
<i>Bilski v. Kappos</i> , 561 U.S. 593 (2010).....	2
<i>Brunfield, Tr. for Ascent Tr. v. IBG LLC</i> , 97 F.4th 854 (Fed. Cir. 2024)	9
<i>Chamberlain Grp., Inc. v. Techtronic Indus. Co.</i> , 935 F.3d 1341 (Fed. Cir. 2019).....	9
<i>ChargePoint, Inc. v. SemaConnect, Inc.</i> , 920 F.3d 759 (Fed. Cir. 2019).....	1
<i>Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat’l Ass’n</i> , 776 F.3d 1343 (Fed. Cir. 2014).....	2, 8
<i>Data Scape Ltd. v. W. Digital Corp.</i> , 816 F. App’x 461 (Fed. Cir. 2020)	10
<i>Elec. Power Grp., LLC v. Alstom S.A.</i> , 830 F.3d 1350 (Fed. Cir. 2016).....	17
<i>Enfish, LLC v. Microsoft Corp.</i> , 822 F.3d 1327 (Fed. Cir. 2016).....	2
<i>Ericsson Inc. v. TCL Commc’n Tech. Holdings Ltd.</i> , 955 F.3d 1317 (Fed. Cir. 2020).....	14, 16
<i>Evolutionary Intel., LLC v. Sprint Nextel Corp.</i> , 137 F. Supp. 3d 1157 (N.D. Cal. 2015), <i>aff’d</i> , 677 F. App’x 679 (Fed. Cir. 2017)	8, 9, 10

<i>Great Plains Tr. Co. v. Morgan Stanley Dean Witter & Co.</i> , 313 F.3d 305 (5th Cir. 2002)	18
<i>Health Discovery Corp. v. Intel Corp.</i> , 577 F. Supp. 3d 570 (W.D. Tex. 2021).....	11
<i>Intell. Ventures I LLC v. Cap. One Bank (USA)</i> , 792 F.3d 1363 (Fed. Cir. 2015).....	7
<i>Prism Techs. LLC v. T-Mobile USA, Inc.</i> , 696 F. App’x 1014 (Fed. Cir. 2017)	9
<i>RecogniCorp, LLC v. Nintendo Co.</i> , 855 F.3d 1322 (Fed. Cir. 2017).....	6
<i>TDE Petroleum Data Sols., Inc. v. AKM Enter., Inc.</i> , 657 F. App’x 991 (Fed. Cir. 2016)	17
<i>Tenstreet, LLC v. DriverReach, LLC</i> , 826 F. App’x 925 (Fed. Cir. 2020)	8
<i>In re TLI Commc’ns LLC Pat. Litig.</i> , 823 F.3d 607 (Fed. Cir. 2016).....	17
Statutes	
35 U.S.C. § 101	1, 2, 3, 12
Federal Rule Of Civil Procedure 12(b)(6)	1, 2

I. INTRODUCTION

An abstract idea is not patentable. Like countless claims that have been invalidated since the Supreme Court’s decision in *Alice Corp. Pty. Ltd. v. CLS Bank Int’l*, 573 U.S. 208 (2014) (“*Alice*”), VirtaMove’s Asserted Patents¹ claim nothing more than using a generic computer system in a conventional manner to replicate and organize data. Thus, they are not directed to patentable subject matter, as required by 35 U.S.C. § 101.

Specifically, the ’814 Patent is directed to the abstract idea of replicating necessary software components into “containers” so they can be used in different environments. And the ’058 Patent is directed to the equally abstract idea of copying shared resources so they can be used by different entities for different purposes at the same time. Both patents recite generic, well-known, prior art components (such as processors and operating systems) that are used in a conventional manner. There is no technological innovation or “inventive concept” that might save either patent. Courts routinely strike down patent claims directed toward such data organization as patent ineligible abstract ideas, and this case should be no exception.

Because this motion raises a pure question of law and no claim construction is required, the Court should dismiss this case with prejudice.

II. LEGAL STANDARDS

The Federal Circuit has consistently held that “[s]ubject matter eligibility under § 101 may be determined at the Rule 12(b)(6) stage of a case.” *ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 765 (Fed. Cir. 2019); *see also Affinity Labs of Tex., LLC v. DIRECTV, LLC*, 838 F.3d 1253 (Fed. Cir. 2016) (affirming district court’s subject matter ineligibility determination at Rule

¹ The “Asserted Patents” are U.S. Patent Nos. 7,519,814 (“’814 Patent”) and 7,784,058 (“’058 Patent”).

12(b)(6) stage). “[P]atent eligibility can be determined at the Rule 12(b)(6) stage” “when there are no factual allegations that, taken as true, prevent resolving the eligibility question as a matter of law.” *Aatrix Software, Inc. v. Green Shades Software, Inc.*, 882 F.3d 1121, 1125 (Fed. Cir. 2018). To determine eligibility at the pleading stage, a court need only have a “full understanding of the basic character of the claimed subject matter”; claim construction is not required. *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat’l Ass’n*, 776 F.3d 1343, 1349 (Fed. Cir. 2014).

“Section 101 of the Patent Act defines the subject matter eligible for patent protection.” *Alice*, 573 U.S. at 216. As the Supreme Court has held, “[l]aws of nature, natural phenomena, and abstract ideas are not patentable.” *Id.* (quoting *Ass’n for Molecular Pathology v. Myriad Genetics, Inc.*, 569 U.S. 576, 589 (2013)). Whether a patent covers patentable subject matter under § 101 is a “threshold” question of law. *Bilski v. Kappos*, 561 U.S. 593, 602 (2010).

In *Alice*, the Supreme Court set forth a two-step framework for determining patent eligibility under Section 101. 573 U.S. at 217-18. **First**, the court determines whether the claims are directed to an “abstract idea[.]” *Id.* at 217. One way to conduct this analysis is “to compare claims at issue to those claims already found to be directed to an abstract idea in previous cases.” *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1334 (Fed. Cir. 2016).

If the claims are directed to an abstract idea, then, **second**, the court “consider[s] the elements of each claim both individually and ‘as an ordered combination’ to determine whether the additional elements ‘transform the nature of the claim’ into a patent-eligible application.” *Alice*, 573 U.S. at 217 (quoting *Mayo Collaborative Servs. v. Prometheus Lab’ys, Inc.*, 566 U.S. 66, 78, 79 (2012)). The Supreme Court has described this as a “search for an ‘inventive concept’—*i.e.*, an element or combination of elements that is ‘sufficient to ensure that the patent in practice amounts

to significantly more than a patent upon the ineligible concept itself.” *Id.* at 217-18 (citations omitted). Implementing an abstract process through generic, conventional computer components and functionality “cannot transform a patent-ineligible abstract idea into a patent-eligible invention.” *Id.* at 223.

III. ARGUMENT

A. The ’814 Patent Claims Are Invalid For Claiming Ineligible Subject Matter

The ’814 Patent is invalid under Section 101 because it fails *Alice*’s two-step test. It fails *Step One* because the claims are directed to the abstract idea of replicating software components into “containers” so they can be used in different environments. Courts have routinely found similar claims unpatentable. It fails *Step Two* because it recites conventional computer components and does not add an inventive concept.

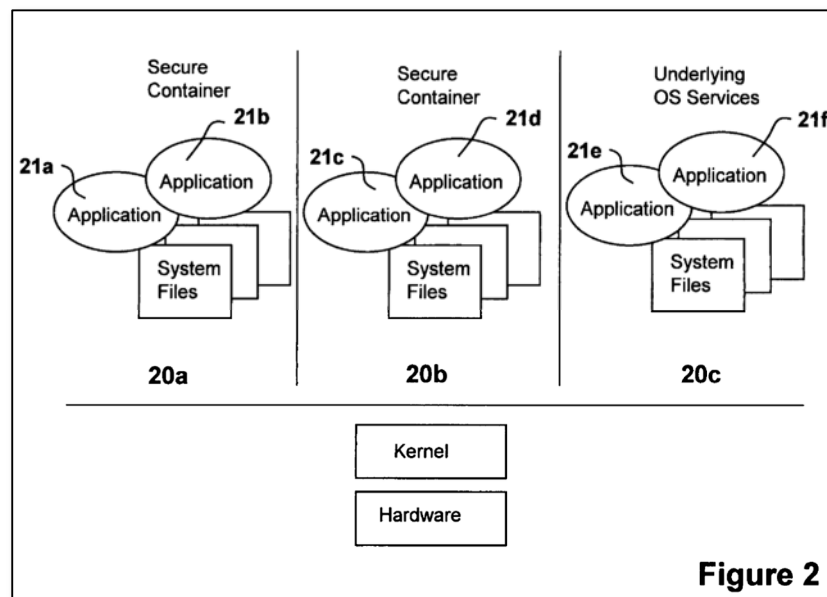
1. Background

The ’814 Patent is entitled “System for Containerization of Application Sets,” but the sole claim specifically asserted in the complaint—claim 1—is a method claim with a single “storing” step. ’814 Patent, 17: 41-61. Thus, just as VirtaMove did in its Complaint, Oracle treats that claim as representative. Although the claim is wordy, the alleged invention is quite simple.

The ’814 Patent describes a way of organizing data on a computer system into what the patent calls “containers.” Notably, the patent does not purport to have invented containers. *See, e.g.*, ’814 Patent, 1:51-2:12 (“Background of the Invention” describing “existing solutions” including containers and identifying prior art product that provides “separation of an application from the underlying operating system.”).

As the patent explains, a container is simply “[a]n aggregate of files required to successfully execute a set of software applications on a computing platform.” ’814 Patent, 2:23-25. In other words, while the computer system has shared hardware (processors, memory, etc.),

“underlying OS [operating system] services” (software managing the system’s shared resources), and a kernel (a privileged part of the operating system capable of directly controlling the hardware), each container has the “application” and “[operating] system files” necessary for that application to execute. That way, the application in the container does not depend on files in the computing system’s operating system because each container contains “all files required to successfully execute a set of software applications on a computing platform.” *Id.*, 7:26-27. Because these system files are now copied and stored in the containers, the files are organized in the container’s “root file system.” *Id.*, 11:35-39. Figure 2 illustrates a system with “containers” of applications and the associated files installed on a server with the underlying kernel and hardware:



The '814 Patent admits that before it was filed, “[c]omputer systems [were] designed in such a way that application programs share common resources.” ’814 Patent, 1:20-21. Thus, the patent concedes “[t]here are existing solutions that address the single use nature of computer systems.” *Id.*, 1:51-52. The patent claims that the “key difference” between the prior art and the purported invention “is that in the former an operating system, including files and a kernel, must be deployed for each application while the latter only requires one operating system regardless of

the number of application containers deployed.” *Id.* 1:56-61. This is nothing more than the abstract concept of replicating and organizing items in an obvious and convenient manner.

2. The '814 Patent Is Directed To An Abstract Idea

Representative claim 1 of the '814 Patent claims the abstract idea of replicating and organizing data into containers so applications can be used in different operating environments.

1. In a system having a plurality of servers with operating systems that differ, operating in disparate computing environments, wherein each server includes a processor and an operating system including a kernel a set of associated local system files compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service, wherein the applications are executed in a secure environment, wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising:

storing in memory accessible to at least some of the servers a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers;

wherein the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems, the containers of application software excluding a kernel,

wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server,

wherein said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server, and

wherein the application software cannot be shared between the plurality of secure containers of application software, and

wherein each of the containers has a unique root file system that is different from an operating system's root file system.

'814 Patent, 17:30-61. To the extent the preamble is limiting, it merely recites generic computing components such as a “server” with “a processor and an operating system” capable of executing an application in a container. There is only one step in this method claim (highlighted yellow),

which is storing containers that include an application and the files required to execute that application. As a whole, representative claim 1 is written in highly generic terms, focusing on “an abstract end-result,” not “‘a specific means or method’ for improving technology.” *RecogniCorp, LLC v. Nintendo Co.*, 855 F.3d 1322, 1326 (Fed. Cir. 2017) (citation omitted).

The abstract nature of representative claim 1 can be illustrated by way of an analogy to a shopping mall food court that offers different cuisines (e.g., pizza, sushi, etc.). The food court might have a single, large kitchen/pantry that contains all of the equipment and ingredients used to make all the styles of food on the menu. That kitchen could employ different chefs (a pizza cook, sushi chef, etc.) who each prepare their style of food, using a single shared stove, shared cookware, and shared ingredients. This is similar to the admitted prior art where the computer and its operating system (the large kitchen) must contain all of the resources (the ingredients and equipment) for each application (chef) to execute (make their style of food).

This approach has obvious problems. For example, chefs would compete for shared resources or there might be cross-contamination problems. An obvious solution is to have separate food stalls, each specializing in its own cuisine, where each chef would each have what they need (and only what they need) to separately prepare their own food. While each food stall might have some of the same condiments (salt and pepper shakers), each stall could also store and serve its own modified versions (parmesan cheese at one stall, soy sauce at another).

In the parlance of the ’814 Patent, a “container” is the food stall and an “application” is the chef. ’814 Patent, 17:42. While there is a pantry area with commonly used ingredients (“associated local system files that remain resident on the server”), each stall has its own set of necessary ingredients and equipment (the “set of associated system files required to execute the one or more applications”). The chef uses the local materials in the food stall rather than running

back and forth to the central storage (“wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server”). The stall’s equipment and ingredients may be the same as what is in the central storage or specific to the stall’s cuisine (“said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server”). To prevent cross-contamination, chefs do not share ingredients or equipment (“the application software cannot be shared between the plurality of secure containers of application software”), and each chef organizes their stall’s ingredients and equipment as they choose (“each of the containers has a unique root file system that is different from an operating system’s root file system”).

While representative claim 1 recites computer-specific terminology such as “applications” and “containers” instead of chefs and stalls, “limiting the use of an abstract idea ‘to a particular technological environment’” cannot render the claims patent eligible. *Alice*, 573 U.S. at 223 (citation omitted). The claims are directed to nothing more than the abstract idea of replicating and organizing data and applications into containers that each include whatever is necessary for that application to operate.

In performing this *Step One* analysis, courts often compare a claim to “the ideas found to be abstract in other cases before the Supreme Court and [the Federal Circuit].” *Intell. Ventures I LLC v. Cap. One Bank (USA)*, 792 F.3d 1363, 1367 (Fed. Cir. 2015). This is instructive here because courts have repeatedly held schemes that are similar to the ’814 Patent to be abstract ideas. Here, the single method step recited in claim 1 is “storing in memory... a plurality of secure containers of application software.” But merely storing data in a computer memory in a particular format is not patent eligible. *See, e.g., Berkheimer v. HP Inc.*, 881 F.3d 1360, 1366 (Fed. Cir.

2018) (claims directed to “parsing, comparing, storing, and editing data” not patent-eligible); *Tenstreet, LLC v. DriverReach, LLC*, 826 F. App’x 925, 926 (Fed. Cir. 2020) (claims directed to “collecting, organizing, and storing data on a conventional computer network” not patent eligible); *Content Extraction*, 776 F.3d at 1345 (claims directed to “receiving” data, “recognizing” portions of that data, and “storing that information in a memory” not patent eligible).

It should be no surprise, then, that patents pertaining to organizing computer information into “containers” have routinely been invalidated as being directed to abstract ideas. For example, in *Hewlett Packard Co. v. ServiceNow, Inc.*, the patent-at-issue claimed “code for creating a hierarchy of derived containers, wherein a given derived container corresponds to,” among other things, “a container definition node of an information model.” No. 14-cv-00570-BLF, 2015 WL 1133244, at *2 (N.D. Cal. Mar. 10, 2015). The Court held that because “the container definition nodes and derived containers are nothing more than a data structure containing information for accessing the information repository hierarchically and a data structure for using that information,” they were directed to the abstract idea of “categorizing and organizing information into a hierarchy.” *Id.* at *7, *9. Here, too, claim 1 of the ’814 Patent attempts to claim a data structure containing information, namely, a “container[] of application software” along with the system files on which the applications depend.

Similarly, in *Evolutionary Intelligence LLC v. Sprint Nextel Corp.*, the Federal Circuit invalidated claims from two patents directed to containers. 677 F. App’x 679, 680 (Fed. Cir. 2017) (“*Evolutionary II*”). The first patent recited a “computer-implemented method” including “searching... first container registers encapsulated and logically defined in a plurality of containers.” *Evolutionary Intel., LLC v. Sprint Nextel Corp.*, 137 F. Supp. 3d 1157, 1161 (N.D. Cal. 2015) (“*Evolutionary I*”), *aff’d*, 677 F. App’x 679 (Fed. Cir. 2017). The second patent recited

“[a]n apparatus... including a plurality of containers, each container being a logically defined data enclosure” comprising various components. *Id.* at 1161-62. The Federal Circuit affirmed the district court’s determination that the claims were invalid for being “directed to the abstract idea of ‘searching and processing containerized data.’” *Evolutionary II*, 677 F. App’x at 680.

The claims of the ’814 Patent are directed to “storing in memory... a plurality of secure containers” with various properties. As in *Evolutionary I*, those properties “are simply functional descriptions of conventional concepts of data processing... to govern the interaction of various data.” *Evolutionary I*, 137 F. Supp. 3d at 1168. That the containers are allegedly “secure” does not make the claim less abstract. To the extent the claim provides for “restricted access to resources,” that is still an abstract idea ineligible for patent protection. *Prism Techs. LLC v. T-Mobile USA, Inc.*, 696 F. App’x 1014, 1017 (Fed. Cir. 2017) (idea of receiving identity data and permitting access only if authorized is abstract).

3. The Asserted Claim Adds Nothing Inventive

Because claim 1 is directed to an abstract idea, it is unpatentable unless in *Alice Step Two* the Court determines the claim adds a “sufficient” inventive concept that “transform[s] the abstract idea... into a patent-eligible application.” *Chamberlain Grp., Inc. v. Techtronic Indus. Co.*, 935 F.3d 1341, 1348-49 (Fed. Cir. 2019). It does not. Representative claim 1 merely recites generic, conventional computer components—for example, servers, operating systems, a processor, a kernel, local system files, applications, and objects—used to achieve the abstract idea of replicating applications into containers so they can be used in different operating environments. “The abstract idea itself cannot supply the inventive concept, no matter how groundbreaking the advance.” *Brumfield, Tr. for Ascent Tr. v. IBG LLC*, 97 F.4th 854, 868 (Fed. Cir. 2024) (citation omitted).

The ’814 Patent does not contend that any of these components is individually novel, nor can it; these components were well known before the ’814 Patent was filed. And each recited

component performs its conventional function. Thus, the claimed computing functions and components, viewed individually and as an ordered combination, comprise the type of “basic functions of a computer” and “purely functional and generic” computer components that cannot supply an inventive concept. *Alice*, 573 U.S. at 225-26.

The structure of the claim demonstrates the lack of an inventive concept. Despite a lengthy preamble, the claim is directed to “a method” with a single step: “storing in memory... a plurality of secure containers of application software.” But the Federal Circuit has held that “storing... data” in memory is “merely us[ing] [a] computer[] for [its] standard function[.]” *Data Scape Ltd. v. W. Digital Corp.*, 816 F. App’x 461, 464 (Fed. Cir. 2020).

Further confirming the lack of inventive concept are cases holding that organizing data into “containers” does not add an inventive concept. In *Hewlett Packard*, the Court rejected claims directed to “container definition nodes” and “derived containers,” explaining that “there is no inventive concept in combining computer readable media with the idea of categorizing and organizing information hierarchically.” 2015 WL 1133244, at *9. And in *Evolutionary I*, the Court found the claims to “merely recite routine and conventional computer operations and structures as a means of implementing the abstract idea of searching and processing containerized data.” 137 F. Supp. 3d at 1169.

Finally, the fact that the containers purport to be “secure” does not transform the abstract idea into patentable subject matter. In *Free Stream Media Corp. v. Alphonso Inc.*, the challenged patents claimed conventional computing devices that “constrain an executable environment in a security sandbox, and execute a sandboxed application in the executable environment.” 996 F.3d 1355, 1358-59 (Fed. Cir. 2021). A “security sandbox,” like the “secure containers” claimed here, “is a security mechanism for separating running programs.” *Id.* at 1359. The Federal Circuit

rejected the patentee’s argument that operating within this “sandbox” was a “specific improvement to the way computers operate.” *Id.* at 1362 (citation omitted). Although the patentee had asserted that “its invention allows devices on the same network to communicate where such devices were previously unable to do so,” the Court rejected that purported technical improvement because, like here, the claim “fail[ed] to recite a practical way of applying [the] underlying idea.” *Id.* at 1363 (quoting *Interval Licensing LLC v. AOL, Inc.*, 896 F.3d 1335, 1343 (Fed. Cir. 2018)).

4. The Remaining Claims Are Not Patent Eligible

The Complaint specifically asserts infringement for only claim 1 of each Asserted Patent and does not allege that claims 2-34 provide a different or additional inventive concept. Oracle thus treats claim 1 as representative for purposes of this Motion. *Health Discovery Corp. v. Intel Corp.*, 577 F. Supp. 3d 570, 577 (W.D. Tex. 2021) (“A district court may analyze representative claims for patent eligibility where all of the asserted and challenged claims are substantially similar and linked to the same purported abstract idea.” (citing *Content Extraction*, 776 F.3d at 1348; *Cleveland Clinic Found. v. True Health Diagnostics LLC*, 859 F.3d 1352, 1360 (Fed. Cir. 2017))); *Berkheimer*, 881 F.3d at 1365 (“Courts may treat a claim as representative in certain situations, such as if the patentee does not present any meaningful argument for the distinctive significance of any claim limitations not found in the representative claim[.]”).² Because all claims are directed to the same abstract idea and none of the claims includes additional elements that transform that abstract idea into patentable subject matter, all claims of the ’814 Patent are unpatentable for the same reason as representative claim 1.

² Should VirtaMove attempt to introduce claims not specifically asserted in the Complaint in opposition to this motion, Oracle reserves the right to address those arguments in its reply.

B. The '058 Patent Claims Are Invalid For Claiming Ineligible Subject Matter

Like the '814 Patent, the '058 Patent is invalid under Section 101. It fails *Step One* because the claims are directed to the abstract idea of replicating and organizing shared resources so they can be used by different entities for different purposes at the same time, and courts have routinely found similar claims unpatentable. It fails *Step Two* because it recites conventional computer components and does not add an inventive concept.

1. Background

The '058 Patent has one independent claim, which is the only claim specifically asserted in the complaint. Thus, Oracle treats claim 1 as representative for purposes of this Motion.

The claimed computing system includes a conventional prior art processor, operating system ("OS"), and shared library. The OS includes "critical system elements" ("CSEs"). The '058 Patent admits that a CSE is a prior art concept that includes "[a]ny service or part of a service, 'normally' supplied by an operating system, that is critical to the operation of a software application." '058 Patent, 6:6-8 (defining CSEs); *id.*, 1:24-28 (noting the use of CSEs in the prior art).

The '058 Patent contends that because CSEs are ordinarily part of the operating system on which applications run (an "OSCSE"), there may be issues if two different applications attempt to use the same CSE at the same time or require different versions or configurations of the same CSE. '058 Patent, 5:41-6:3. The '058 Patent purports to solve this problem by "enabl[ing] the replication of critical system elements normally found in an operating system kernel. These replicated CSEs are then able to run in the context of a software application." *Id.*, 5:21-24. In essence, rather than using only an OSCSE, the '058 Patent "replicate[s]" the CSE for use by an application or set of applications. *Id.* The copy is called a shared library CSE ("SLCSE"). *Id.*, 2:27-30, 3:23-25.

Like the '814 Patent, the '058 Patent is directed to the abstract idea of replicating and organizing data. This is confirmed in the patent's figures. "Prior Art" FIG. 2b "is illustrative of a known system architecture where critical system elements execute in user mode and execute in distinct context from applications in a single application process context" while "FIG. 3 is an architectural view of an embodiment of the invention." '058 Patent, 5:5-10.

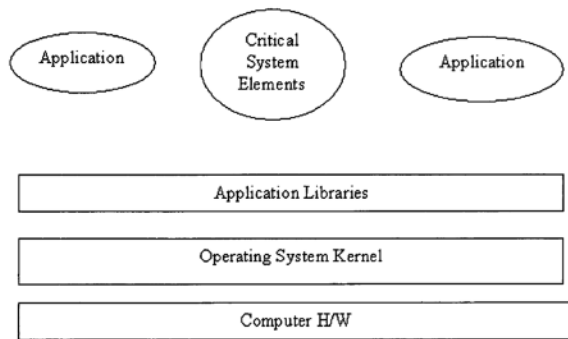


FIG. 2b
Prior Art

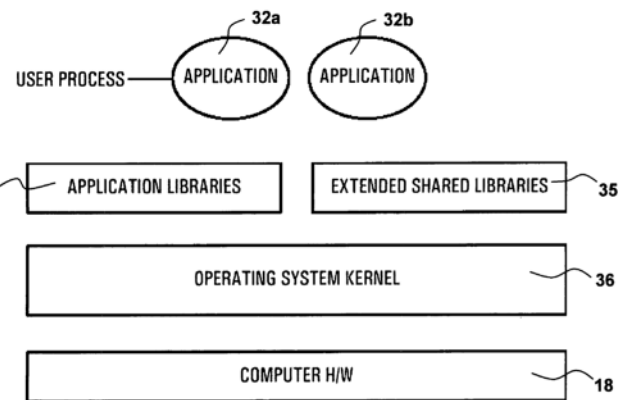
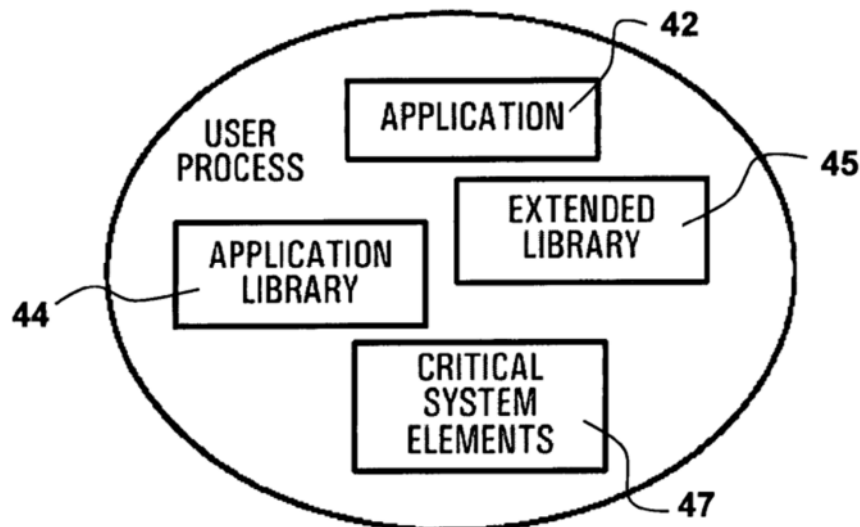


FIG. 3

Both the prior art and the alleged invention include the same components, but the alleged invention replicates and organizes the information. This is shown in the excerpt of FIG. 4, which illustrates the "User Process" (32a, 32b) of FIG. 3 to make clear that the invention merely copied certain prior art components (such as the critical system elements) into the user process:



2. The '058 Patent Is Directed To An Abstract Idea

While “written in technical jargon,” representative claim 1 is directed to the abstract idea of replicating and organizing shared resources so they can be used by different entities for different purposes at the same time. *See Ericsson Inc. v. TCL Commc’n Tech. Holdings Ltd.*, 955 F.3d 1317, 1326 (Fed. Cir. 2020) (“*Ericsson*”).

1. A computing system for executing a plurality of software applications comprising:

- a) a processor;
- b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,
- c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and
 - i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications,
 - ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and
 - iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.

'058 Patent, 10:51-11:14. The claimed “computing system” has only three elements—a conventional processor, operating system, and shared library. The prior art CSEs are replicated and organized to be stored in both the OS (as OSCSEs) and the shared library (as SLCSEs).

Although the claims of the '058 Patent do not use the term “container,” their abstract idea can be illustrated using the same food court analogy described above. In the parlance of the '058 Patent, there is a shared pantry (the “operating system”) with common ingredients (“OS critical system elements (OSCSEs)”). Each food stall, however, also has the specific equipment and ingredients needed to prepare its cuisine (“shared library having shared library critical system elements (SLCSEs)”). And, once again, various food stalls may serve the same things (e.g., salt and pepper) from the shared pantry (“wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications”). But each chef may also use certain equipment or ingredients unique to their restaurant. For example, the Italian food stall would be the only one with a pizza oven and it is not shared with the other food stalls (“wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications”). In this way, each food stall can prepare similar types of food but in unique ways (“at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function”). By replicating common elements (stoves, pots, pans, etc.) in each food stall, the chefs can use that equipment (“SLCSEs”) to prepare food simultaneously (“wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously”).

Like the '814 Patent, the '058 Patent is directed to replicating and organizing data, which, as shown in the cases cited in § III.A.2, courts have consistently found to be an abstract idea. If anything, the '058 Patent is more abstract than the '814 Patent because the '058 Patent does not even purport to require “containers” or some other specific organizational structure.

The Federal Circuit’s decision in *Ericsson* makes clear that the technical jargon in the '058 Patent does not save the patent from being directed to an abstract idea. 955 F.3d at 1326. The *Ericsson* patent claimed “a method and system for limiting and controlling access to resources in a telecommunications system.” *Id.* at 1320. The Federal Circuit held that, although the claims were “written in technical jargon,” they were nevertheless “directed to the abstract idea of controlling access to, or limiting permission to, resources.” *Id.* at 1326. The Court rejected the patentee’s contention that the claims “solve the specific computer problem... of controlling app access in resource-constrained mobile phones” because the claims did not “ha[ve] the specificity required to transform a claim from one claiming only a result to one claiming a way of achieving it.” *Id.* at 1327-28 (citation omitted). Here, as in *Ericsson*, representative claim 1 is jargon rich, but merely recites conventional components (a processor, operating system, and a shared library), and three “wherein” clauses listing goals without any specific “way of achieving it.” *Id.* at 1328.

Simply put, the problem that the '058 Patent purports to address—the inefficiencies associated with multiple applications attempting to run using the same critical system elements—is age-old and not unique to the computer context. And its alleged solution—to replicate and organize system elements—is an unpatentable abstract idea.

3. The Asserted Claim Adds Nothing Inventive

The '058 Patent fails *Alice Step Two* for reasons similar to the '814 Patent. Claim 1 recites only generic computer elements and functions—“processor,” “operating system[s],” “critical system elements,” “shared librar[ies],” and “applications”—used to achieve the abstract idea. '058

Patent, 10:51-11:14. Neither the '058 Patent nor VirtaMove's Complaint alleges that any of these components was non-conventional, and the '058 Patent acknowledges the components are prior art. For example, FIGs. 1, 2a, and 2b are admitted "Prior Art" that include the claim 1 components.

Although VirtaMove may argue the "ordered combination" of these conventional elements transforms the idea, even a cursory comparison of the prior art (FIGs. 1-2b) and the allegedly inventive figures (FIGs. 3-6) shows that the combination of claimed elements was routine. *See Free Stream Media*, 996 F.3d at 1365; *In re TLI Commc'ns LLC Pat. Litig.*, 823 F.3d 607, 615 (Fed. Cir. 2016) (where "recited physical components behave exactly as expected according to their ordinary use," they do not constitute an inventive concept).

Finally, the structure of claim 1 illustrates that it lacks an inventive concept. As noted, the claim recites just three conventional computer components: (a) "a processor"; (b) "an operating system"; and (c) "a shared library" that are in the admitted prior art. *See* '058 Patent, FIGs 1-2a. The claim then includes three "wherein" clauses that recite a wish list of features of the claimed components. This is insufficient. As the Federal Circuit has repeatedly explained, claims that "merely call for performance of the claimed... functions" without explaining "**how** the desired result is achieved" cannot supply an inventive concept. *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1355 (Fed. Cir. 2016); *TDE Petroleum Data Sols., Inc. v. AKM Enter., Inc.*, 657 F. App'x 991, 993 (Fed. Cir. 2016) (rejecting claims that "recite the what of the invention, but none of the **how** that is necessary to turn the abstract idea into a patent-eligible application") (original emphasis in both). By merely reciting the desired features and functional characteristics in the passive voice (e.g., "are accessible" and "is provided"), claim 1 has not provided the "**how**" required to transform the abstract idea into a patentable invention.

4. The Remaining Claims Are Not Patent Eligible

As with the '814 Patent, claim 1 is the only specifically asserted claim in the Complaint. VirtaMove treats claim 1 as representative and has not identified any allegedly patentable differences in claims 2-18. All claims should be found unpatentable.

IV. CONCLUSION

For the reasons set forth above, Oracle respectfully requests that the Court dismiss VirtaMove's Complaint with prejudice.³

Dated: March 14, 2025

Respectfully submitted,

/s/ Jared Bobrow

Jared Bobrow (admitted *pro hac vice*)

CA Bar No. 133712

jbobrow@orrick.com

Bas de Blank (admitted *pro hac vice*)

CA Bar No. 191487

basdeblank@orrick.com

Diana Rutowski (admitted *pro hac vice*)

CA Bar No. 233878

drutowski@orrick.com

Orrick, Herrington & Sutcliffe LLP

1000 Marsh Road

Menlo Park, CA 94025

Tel: (650) 614-7400

Fax: (650) 614-7401

Darryl J. Adams

Slayden Grubert Beard

401 Congress Avenue, Suite 1650

Austin, Texas 78701

Tel: 512-402-3550

Attorneys for Defendant Oracle Corp.

³ Oracle seeks dismissal with prejudice, as “it is clear that the defects [in VirtaMove’s Complaint] are incurable”—the Asserted Patents are demonstrably invalid, and no amendment VirtaMove may seek to make can alter this fact. *Great Plains Tr. Co. v. Morgan Stanley Dean Witter & Co.*, 313 F.3d 305, 329 (5th Cir. 2002).

CERTIFICATE OF SERVICE

The undersigned certifies that on March 14, 2025, all counsel of record who are deemed to have consented to electronic service are being served with a copy of this document through the Court's CM/ECF system.

/s/ Jared Bobrow

Jared Bobrow